

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : David M. Baggett

Art Unit : not assigned

Continuation of:

Examiner : not assigned.

Serial No. : 09/272,437

Filed : March 19, 1999

Title : TECHNIQUE FOR PRODUCING CONSTRUCTED FARES

Commissioner for Patents

Washington, D.C. 20231

PRELIMINARY AMENDMENT

Prior to examination, please amend the application as follows:

In the specification:

Replace the paragraph beginning at page 5, line 24 with the following rewritten paragraph:

Referring now to FIG. 1, a computer system 10 includes a CPU 12, main memory 14 and persistent storage device 16 all coupled via a computer bus 18. The system 10 also includes output devices such as a display 20 and a printer (not shown), as well as user input devices such as a keyboard 24 and a mouse 26. Not shown in FIG. 1, but would necessarily be included in a system of FIG. 1, are software drivers and hardware interfaces to couple all the aforementioned elements to the CPU 12.

Replace the paragraphs beginning at page 10, line 9, and ending on page 12, line 3 with the following rewritten paragraphs:

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL624272755US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

June 8, 2001

Date of Deposit

Signature

Samantha Bell
Typed or Printed Name of Person Signing Certificate

The fare construction process 40 retrieves 44 an interior city from a list of interior cities "C1" that are part of an arbitrary for the airline "A." The fare construction process 40 retrieves 46 a city C2 from the list 38a (via an access to the first hash table 37a) of all of the cities "C2" that form arbitraries C1-C2 for city "C1" for airline "A." The constructed fares process 30 retrieves 48 from the list 38b (via an access to the second hash table 38a) a city "C3" that forms a published fare with city "C2" for airline "A." The fare construction process 40 tries 50 to produce a constructed fare for the C1-C2-C3 city combinations. This embodiment of the fare construction process 40 produces the two-component constructed fares, i.e., one arbitrary combined with one published fare.

The fare construction process 40 will determine 52 if the C1-C2-C3 constructed fare is valid. If it is valid it is stored 54 or otherwise noted, and then a pointer (not shown) is incremented to get the next C3 city 56. If there are more C3 cities 58, the fare construction process will retrieve 48 the next gateway city C3 for which a published fare exists with C2 that is all of the cities that form a published fare with gateway city C2 and the fare construction process 40 will continue. If all cities C3 have been tested, it will then retrieve the next C2 city 60 by incrementing another pointer (not shown) to get the next C2 city. If there are more C2 cities 62, the fare construction process 40 will retrieve 46 the next gateway city C2 for which an arbitrary exists with city C1 and the fare construction process 40 will continue.

If all C2 cities have been tested, it will then retrieve the next C1 city 64 by incrementing another pointer (not shown) to get the next C1 city 44. If there are more C1 cities, the fare construction process will retrieve the next interior city C1 will continue. If all interior cities C1 have been tested, it will then retrieve the next airline 66 by incrementing another pointer (not shown) to get the next airline.

Referring now to FIGS. 4A-4C, a fare construction process 40' to construct two and three-component constructed fares, i.e., two arbitraries combined with one published fare is shown. In the embodiment 40', there is an additional loop. The fare construction process 40' is similar to the process 40 up to when the fare construction process tries 50 to construct a two-component constructed fare. Thus, the fare construction process 40' uses processes 42, 44, 46, 48 as well as 52, 54, 56, 58 and 60, 62, 64, 66 and 68 of FIGS. 3A-3B. The fare construction process 40' retrieves 51a a second interior city C4 from a list of interior cities "C4" that are part

of an arbitrary that exists for city C3 for the airline "A." The fare construction process 40 retrieves 51a a city C4 from the list 38c (via an access to the third hash table 37c) of all of the cities "C4" that form arbitraries C3-C4 for gateway city "C3" for airline "A." The fare construction process 40' tries 51b to produce a constructed fare for the C1-C2-C3-C4 city combinations. The fare construction process 40 will determine 51c if the C1-C2-C3-C4 constructed fare is valid. If it is valid it is stored 54a or otherwise noted, and then a pointer (not shown) is incremented to get 51d the next C4 city. If there are more C4 interior cities 51e, the fare construction process 40' will retrieve the next interior city C4 and the fare construction process will continue. If all cities C4 have been tested, it will continue from testing 52 the C1-C2-C3 two component constructed fare as described above. This embodiment 40' produces two-component and three-component constructed fares, i.e., two arbitraries combined with one published fare.

Replace the paragraphs beginning at page 14, line 15, and ending on page 15, line 8 with the following rewritten paragraphs:

Referring to FIG. 5, in determining whether or not a fare can combine with an arbitrary 94, the fare construction process 40 checks the "PO2" records corresponding to the fare 96. If the PO2 records do not list the arbitrary's fare code, the arbitrary cannot combine with the fare, and the construction is prohibited. Otherwise, the fare construction process 40 returns a pass 98. The conventional process of finding the set of PO2 records matching a particular fare is time consuming. Alternatively, the check of fare construction tables can be done on a batch basis after all potential constructed fare combinations have been determined.

To save time when constructing large numbers of fares, a PO2 procedure retrieval 80 uses a process called memoization. Memoization is a technique for speeding up certain kinds of algorithms. If an expensive procedure is called many times, and if the procedure's output depends only on the input (i.e., the answer is not dependent on any external factors, such as the current time), then memoization can be used. To memoize PO2 retrieval, the PO2 procedure call 80 forms a query 82 involving the constructed fare. The memoization retrieval process 80 has a store 88 of past queries and associated answers. If the memoization retrieval process 80 determines 84 that the query has been stored, it is retrieved from the store 92. Otherwise, a procedure call 86 for the PO2 record is produced and used to access 90 the record from a remote

database. The answer from the remote database is stored 92 in the memoization store 88 for future references.

In the claims:

Cancel claim 47-51.

Amend claims 1, 2, 4, 5, 11, 14, as follows:

(Amended) 1. A method of producing constructed fares that includes an arbitrary added to a published fare, said method executed in a computer system, the method comprising:

determining interior cities that appear with gateway cities in arbitraries for the airline, the arbitraries extending published fares, in order to derive prices to minor cities;

searching a database having published fares for gateway cities corresponding to the determined interior cities appearing in the arbitraries; and

applying an arbitrary corresponding to one of the interior cities to a published fare involving one of the gateway cities determined interior cities appearing in the arbitraries to produce the constructed fare.

(Amended) 2. The method of claim 1 wherein determining interior cities comprises:

accessing a hash table indexed by an airline, interior-city pair to return a list of gateway cities for which an airline has arbitraries that specify the interior city.

(Amended) 4. The method of claim 1 wherein searching for gateway cities comprises:

accessing a hash table indexed by an airline, gateway pair to return a list of gateway cities that an airline publishes fares from the determined gateway to another gateway city.

(Amended) 5. The method of claim 1 wherein determining interior cities comprises:

accessing a first hash table indexed by an airline, interior-city pair to return a list of gateway cities for which an airline has arbitraries that specify the interior city; and wherein searching for gateway cities further comprises:

accessing a second hash table indexed by an airline, gateway pair to return a second list of gateway cities that an airline publishes fares from the determined gateway to another gateway city.

(Amended) 11. The method of claim 1, further comprising:

determining a second set of interior cities that appear with a second gateway city in the published fare for the airline;

applying an arbitrary that extends the published fare to a city from the second set of interior cities to produce a three component constructed fare.

(Amended) 14. A method of producing a set of constructed international fares for a particular airline, the method executed in a computer system and the method comprising:

determining interior cities that appear with gateway cities in arbitraries for the particular airline;

searching for gateway cities corresponding to the determined interior cities appearing in the arbitraries; and

applying arbitraries corresponding to the determined interior cities to published fares involving the gateway cities to produce the set of constructed fares for the airline.

Please add claims 52-55, as follows:

52. A method of producing a set of constructed international fares for an airline, the method executed in a computer system and the method comprising:

accessing a first hash table by airline interior city pair to return a list of gateway cities for which an airline has arbitraries that specify the interior city;

accessing a second hash table by airline gateway pair to return a second list of gateway cities that an airline publishes fares from to determine a gateway to another gateway city;

applying the first arbitraries from the first hash table to the published fares from the second hash table to return a list of potential constructed fares; and

determining whether a constructed fare in a list of constructed fares is a valid constructed fare and for valid ones of the potential constructed fares producing the constructed fares.

53. The method of claim 52 wherein the constructed fares are two component constructed fares.

54. The method of claim 52 further comprising:
determining a second set of interior cities that appear as arbitraries with the second gateway cities by accessing a third hash table by airline interior city pair to return a list of gateway cities for which an airline has arbitraries that specify the interior city; and
applying the arbitraries corresponding to the determined second set of interior cities to the first set of constructed fares to produce a second set of constructed fares.

55. The method of claim 54 wherein the first set of constructed fares are two component constructed fares and the second set of constructed fares are three component constructed fares.

In the drawings:

A Request for Approval of Drawing Changes accompanies this response.

In the drawings, Applicant has amended FIG. 1 to show database 20a. Applicant has amended FIG. 2 to add the reference number "30", has amended FIG. 3A to label the hash tables 37a and 37b and show lists 38a and 38b. Applicant has amended FIG. 3B to show the continuation from the store process 54 to retrieving the next C3 city, has added "yes" and "no" designations after each of the decision blocks and has amended FIG. 5 to show the continuation after store steps 58a and 54 and has labeled the decision blocks with "yes" and "no" and showed where the code branches from decision block 51e back to 51a. In FIG. 5 Applicant has added designation 80 and has modified FIG. 5 to show the memorization process 84 occurring before the call fare rule process 86 as consistent with the specification. In the drawings, Applicant has amended FIG. 5 to label decision blocks with "yes" and "no" consistent with the specification.

These changes are supported in the specification and no new matter has been added.

REMARKS

Applicant has filed this continuation application to seek protection on subject matter that was canceled in the parent case. Applicant has made clarifying amendments to certain of the claims to more particularly point out the subject matter of the invention.

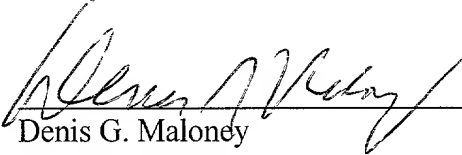
Attached is a marked-up version of the changes being made by the current amendment.

Applicant asks that all claims be examined. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: _____

6/8/2001



Denis G. Maloney
Reg. No. 29,670

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

Version with markings to show changes made

Replace the paragraph beginning at page 5, line 24 with the following rewritten paragraph:

Referring now to FIG. 1, a computer system 10 includes a CPU 12, main memory 14 and persistent storage device 16 all coupled via a computer bus 18. The system 10 also includes output devices such as a display 20 and a printer [22] (not shown), as well as user input devices such as a keyboard 24 and a mouse 26. Not shown in FIG. 1, but would necessarily be included in a system of FIG. 1, are software drivers and hardware interfaces to couple all the aforementioned elements to the CPU 12.

Replace the paragraphs beginning at page 10, line 9, and ending on page 12, line 3 with the following rewritten paragraphs:

The fare construction process 40 retrieves 44 an interior city from a list of interior cities "C1" that are part of an arbitrary for the airline "A." The fare construction process 40 retrieves 46 [44] a city C2 from the list 38a (via an access to the first hash table 37a) of all of the cities "C2" that form arbitraries C1-C2 for city "C1" for airline "A." The constructed fares process 30 retrieves 48 from the list 38b (via an access to the second hash table 38a) a city "C3" that forms a published fare with city "C2" for airline "A." The fare construction process 40 tries 50 to produce a constructed fare for the C1-C2-C3 city combinations. This embodiment of the fare construction process 40 produces the two-component constructed fares, i.e., one arbitrary combined with one published fare.

The fare construction process 40 will determine 52 if the C1-C2-C3 constructed fare is valid. If it is valid it is stored 54 or otherwise noted, and then a pointer (not shown) is incremented to get the next C3 city 56. If there are more C3 cities 58, the fare construction process will retrieve 48 the next gateway city C3 for which a published fare exists with C2 that is all of the cities that form a published fare with gateway city C2 and the fare construction process 40 will continue. If all cities C3 have been tested, it will then retrieve the next C2 city 60 by incrementing another pointer (not shown) to get the next C2 city. If there are more C2 cities 62,

the fare construction process 40 will retrieve 46 the next gateway city C2 for which an arbitrary exists with city C1 and the fare construction process 40 will continue.

If all C2 cities have been tested, it will then retrieve the next C1 city 64 by incrementing another pointer (not shown) to get the next C1 city 44. If there are more C1 cities, the fare construction process will retrieve the next interior city C1 will continue. If all interior cities C1 have been tested, it will then retrieve the next airline 66 by incrementing another pointer (not shown) to get the next airline.

Referring now to FIGS. 4A-4C, a fare construction process 40' to construct two and three-component constructed fares, i.e., two arbitraries combined with one published fare is shown. In the embodiment 40', there is an additional loop. The fare construction process 40' is similar to the process 40 up to when the fare construction process tries 50 to construct a two-component constructed fare. Thus, the fare construction process 40' uses processes 42, 44, 46, 48 as well as 52, 54, 56, 58 and 60, 62, 64, 66 and 68 of FIGS. 3A-3B. The fare construction process 40' retrieves 51a a second interior city C4 from a list of interior cities "C4" that are part of an arbitrary that exists for city C3 for the airline "A." The fare construction process 40 retrieves [44] 51a a city C4 from the list 38c (via an access to the third hash table 37c) of all of the cities "C4" that form arbitraries C3-C4 for gateway city "C3" for airline "A." The fare construction process 40' tries 51b to produce a constructed fare for the C1-C2-C3-C4 city combinations. The fare construction process 40 will determine 51c if the C1-C2-C3-C4 constructed fare is valid. If it is valid it is stored 54a or otherwise noted, and then a pointer (not shown) is incremented to get 51d the next C4 city. If there are more C4 interior cities 51e, the fare construction process 40' will retrieve the next interior city C4 and the fare construction process will continue. If all cities C4 have been tested, it will continue from testing 52 the C1-C2-C3 two component constructed fare as described above. This embodiment 40' produces two-component and three-component constructed fares, i.e., two arbitraries combined with one published fare.

Replace the paragraphs beginning at page 14, line 15, and ending on page 15, line 8 with the following rewritten paragraphs:

[In] Referring to FIG. 5, in determining whether or not a fare can combine with an arbitrary 94, the fare construction process 40 checks [84] the "PO2" records corresponding to the

fare 96. If the PO2 records do not list the arbitrary's fare code, the arbitrary cannot combine with the fare, and the construction is prohibited. Otherwise, the fare construction process 40 returns a pass 98. The conventional process of finding the set of PO2 records matching a particular fare is time consuming. Alternatively, the check of fare construction tables can be done on a batch basis after all potential constructed fare combinations have been determined.

[Referring now to FIG, 5, to] To save time when constructing large numbers of fares, a PO2 procedure retrieval 80 uses a process called memoization. Memoization is a technique for speeding up certain kinds of algorithms. If an expensive procedure is called many times, and if the procedure's output depends only on the input (i.e., the answer is not dependent on any external factors, such as the current time), then memoization can be used. To memoize PO2 retrieval, the PO2 procedure call 80 forms a query 82 involving the constructed fare. The memoization retrieval process 80 has a store 88 of past queries and associated answers. If the memoization retrieval process 80 determines 84 that the query has been stored, it is retrieved from the store 92. Otherwise, a procedure call 86 for the PO2 record is produced and used to access 90 the record from a remote database. The answer from the remote database is stored 92 in the memoization store 88 for future references.

In the claims:

Cancel claim 47-51.

Amend claims 1, 2, 4, 5,11 and 14 as follows:

(Amended) 1. A method of producing constructed fares that includes an arbitrary added to a published fare, said method executed in a computer system, the method comprising:

determining interior cities that appear with gateway cities in arbitraries for the airline, the arbitraries extending published fares, in order to derive prices to minor cities;

searching a database having published fares for gateway cities corresponding to the determined interior cities appearing in the arbitraries; and

applying an arbitrary corresponding to one of the interior cities to a published fare involving one of the gateway cities determined interior cities appearing in the arbitraries to produce the constructed fare.

(Amended) 2. The method of claim 1 wherein determining interior cities comprises:
accessing a hash table indexed by an airline, interior-city pair to return a list of gateway cities for which an airline has arbitraries that specify the interior city.

(Amended) 4. The method of claim 1 wherein searching for gateway cities comprises:
accessing a hash table indexed by an airline, gateway pair to return a list of gateway cities that an airline publishes fares from the determined gateway to another gateway city.

(Amended) 5. The method of claim 1 wherein determining interior cities comprises:
accessing a first hash table indexed by an airline, interior-city pair to return a list of gateway cities for which an airline has arbitraries that specify the interior city; and wherein searching for gateway cities further comprises:

accessing a second hash table indexed by an airline, gateway pair to return a second list of gateway cities that an airline publishes fares from the determined gateway to another gateway city.

(Amended) 11. The method of claim 1, further comprising:
determining a second set of interior cities that appear with a second gateway city in the published fare for the airline;

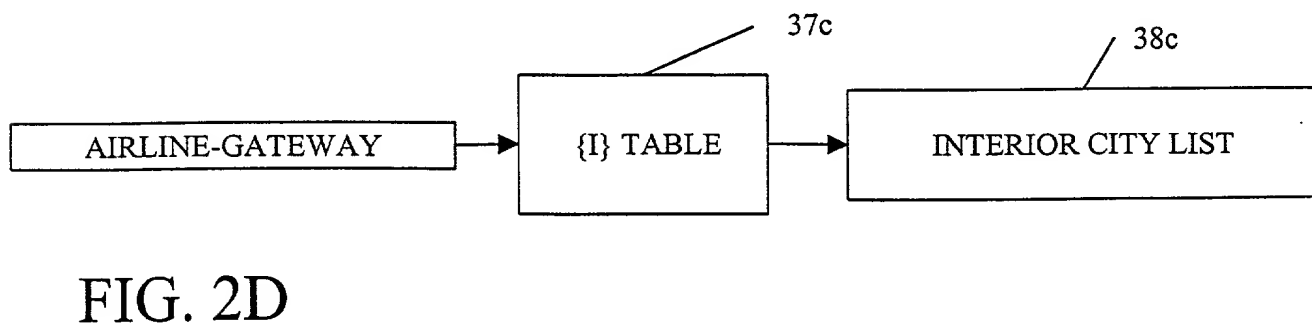
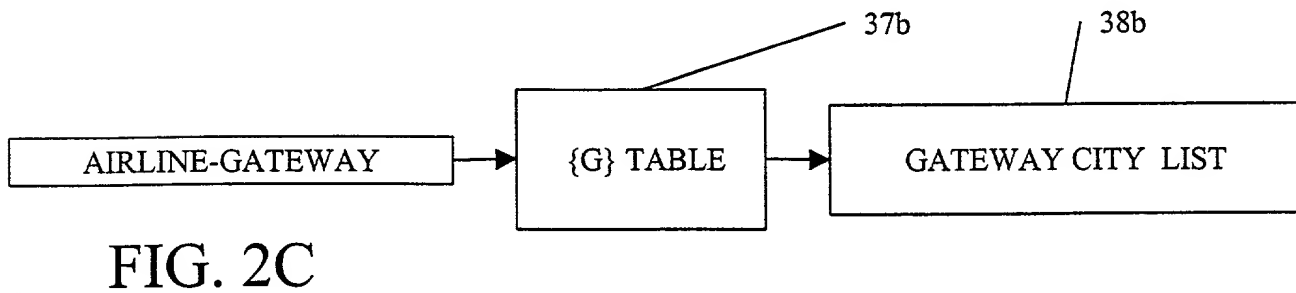
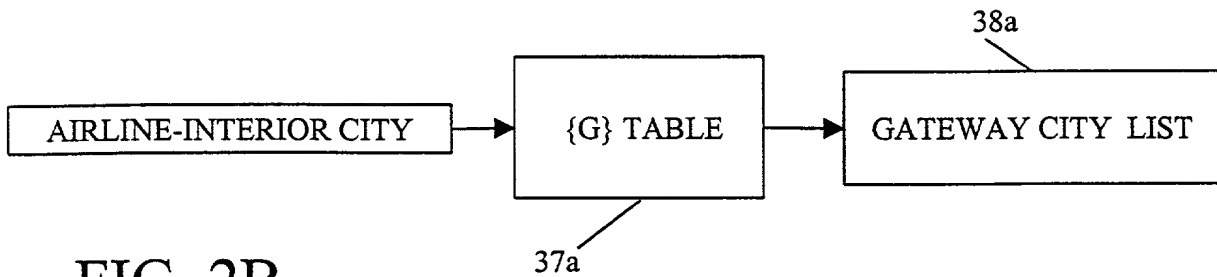
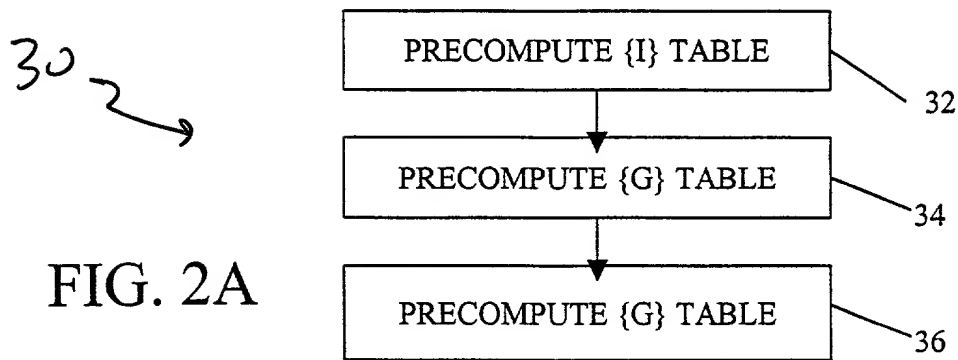
applying an arbitrary that extends the published fare to a city from the second set of interior cities [a second interior city from the second set of interior cities to the constructed fare] to produce a three component constructed fare.

(Amended) 14. A method of producing a set of constructed international fares for [an] a particular airline, the method executed in a computer system and the method comprising:

determining interior cities that appear with gateway cities in arbitraries for the particular airline;

searching for gateway cities corresponding to the determined interior cities appearing in the arbitraries; and

applying [each arbitrary] arbitraries corresponding to [each of] the determined interior cities to published fares involving the gateway cities to produce the set of constructed fares for the airline.



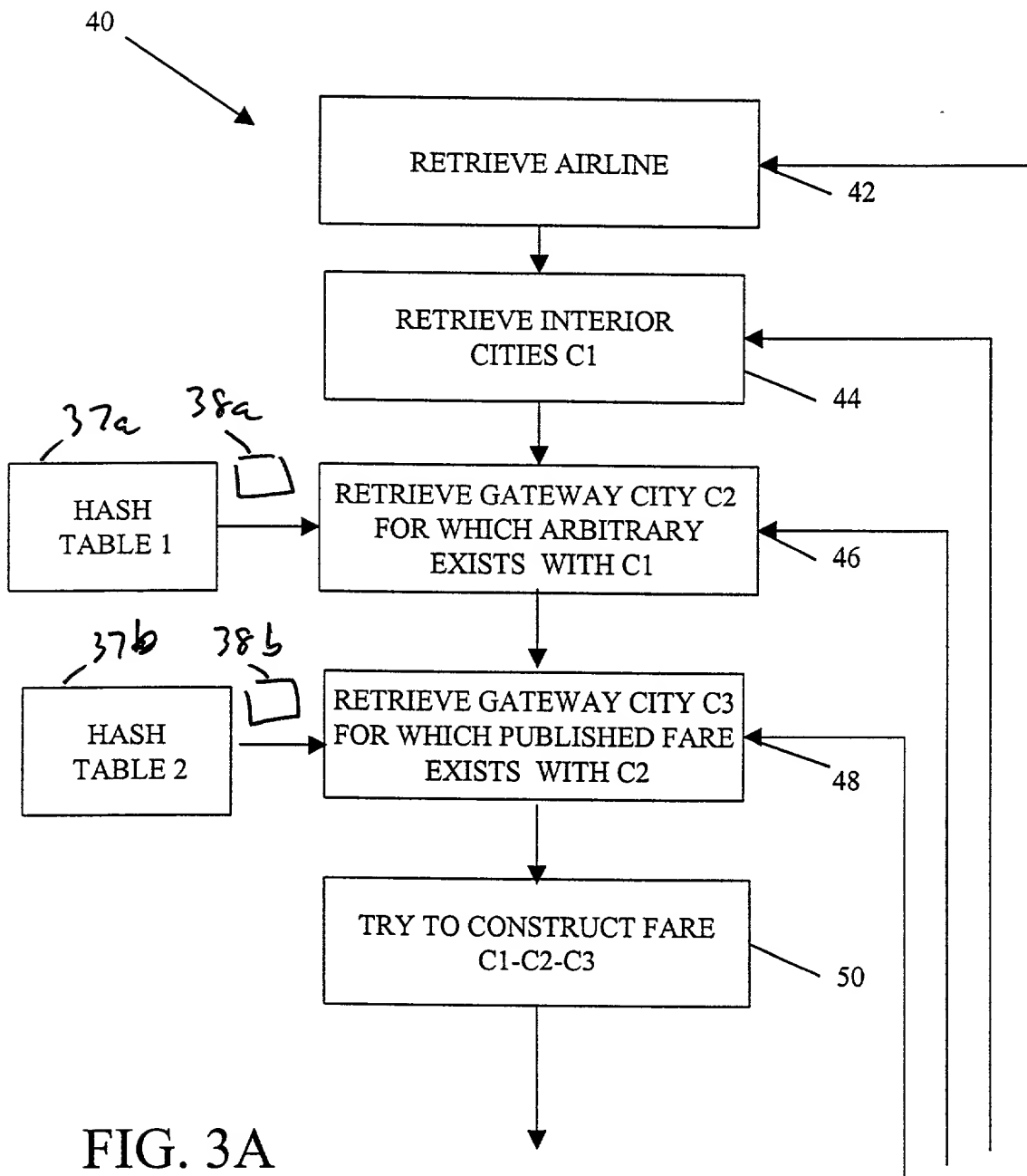


FIG. 3A

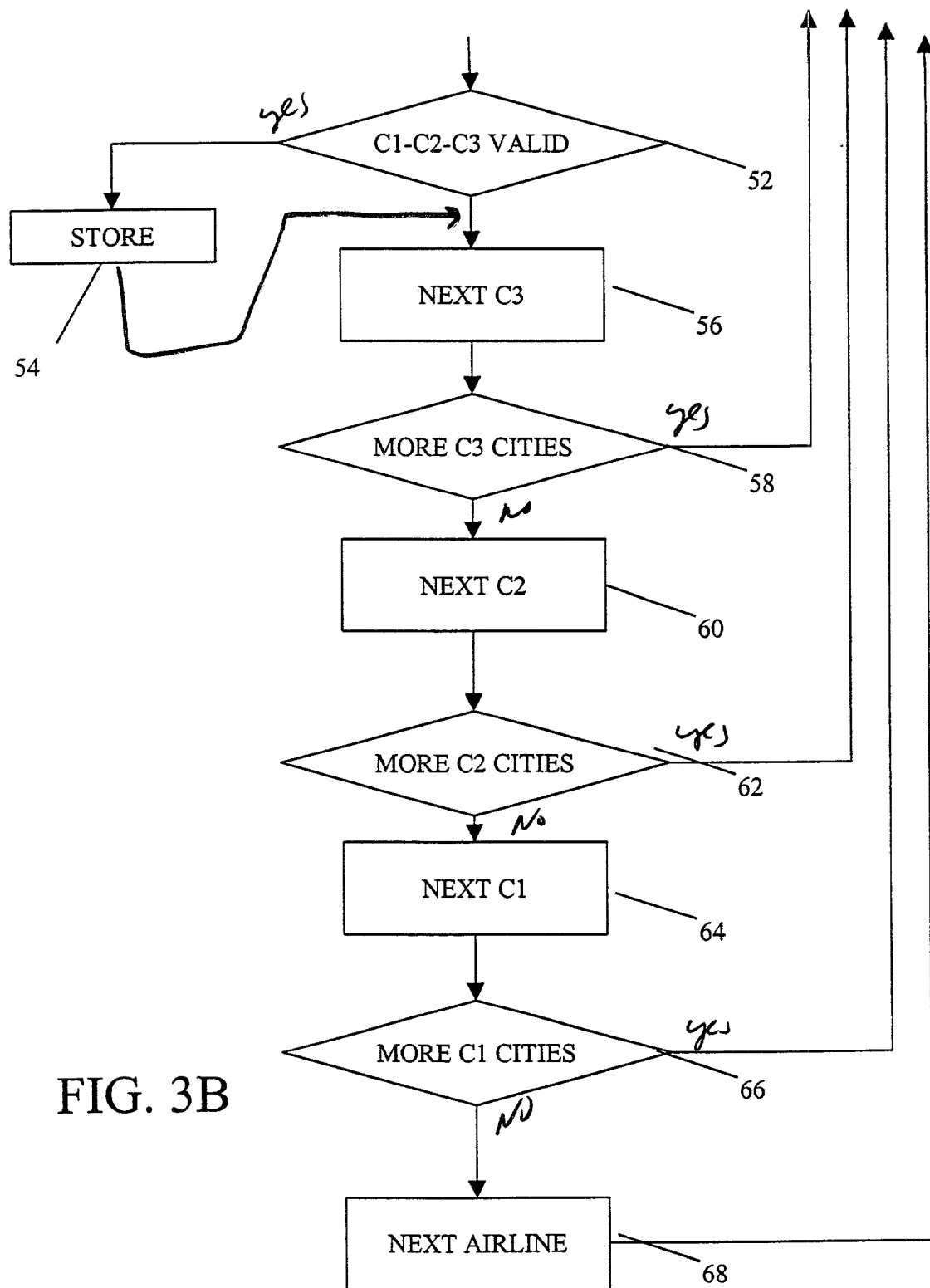
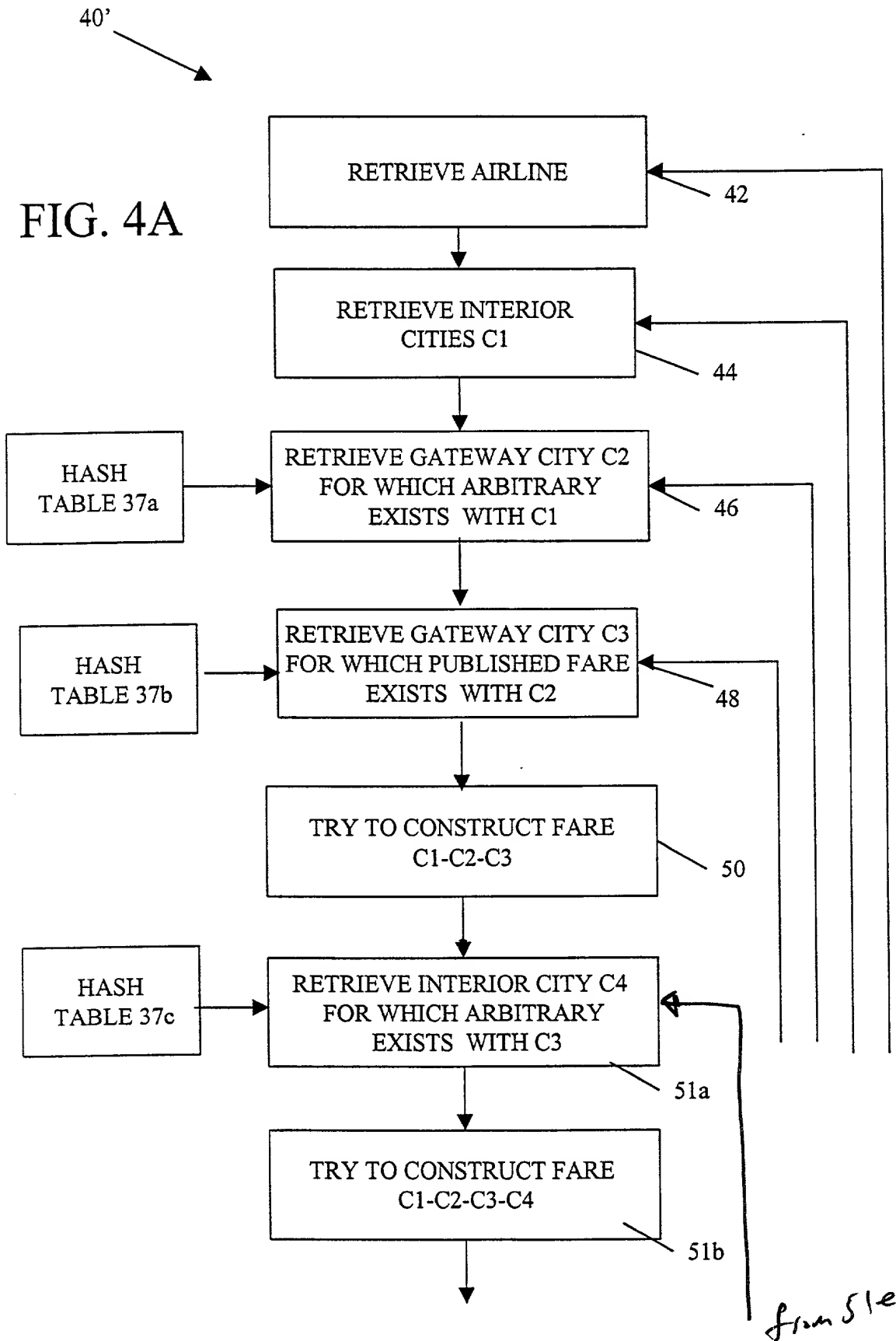


FIG. 4A



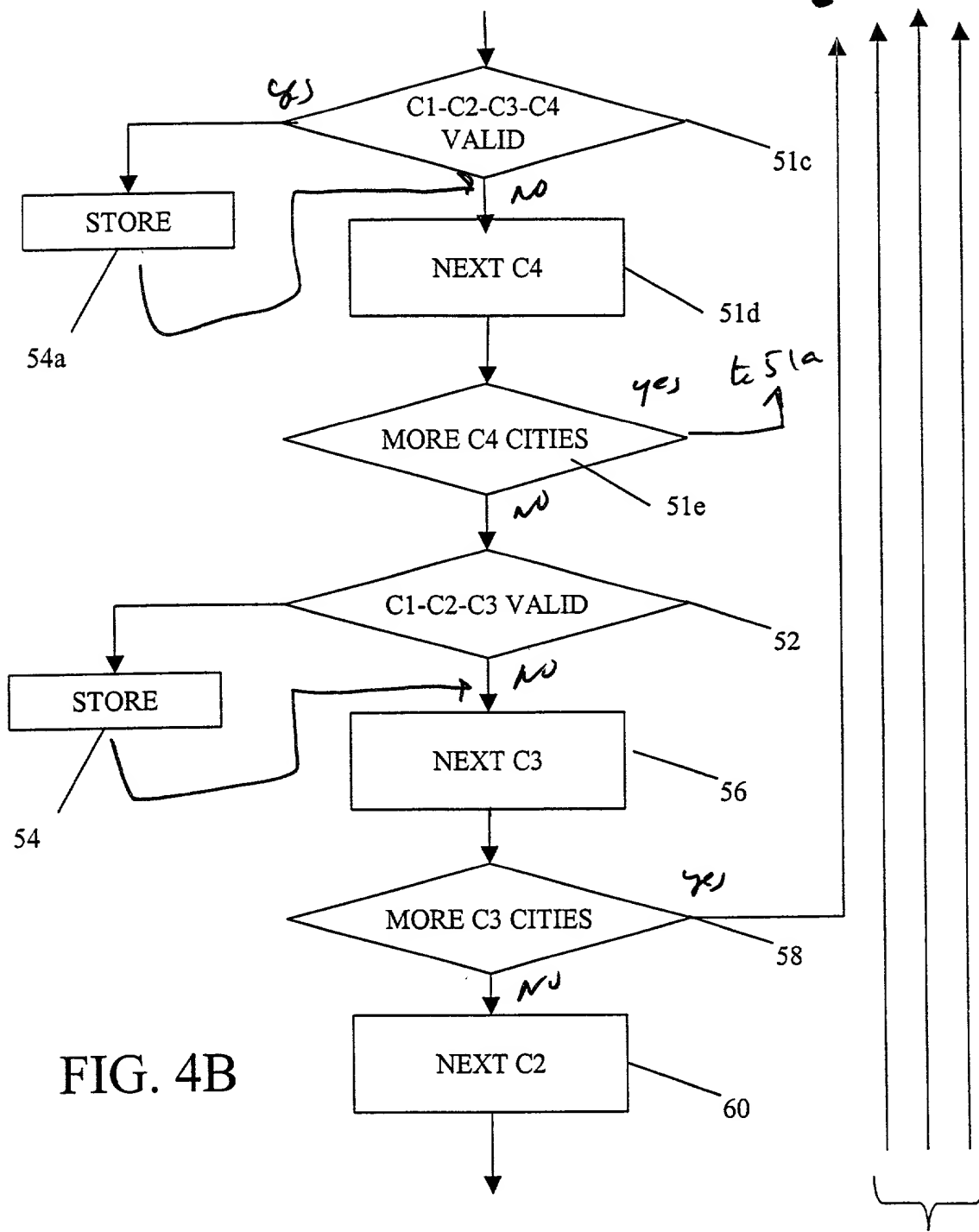


FIG. 4C

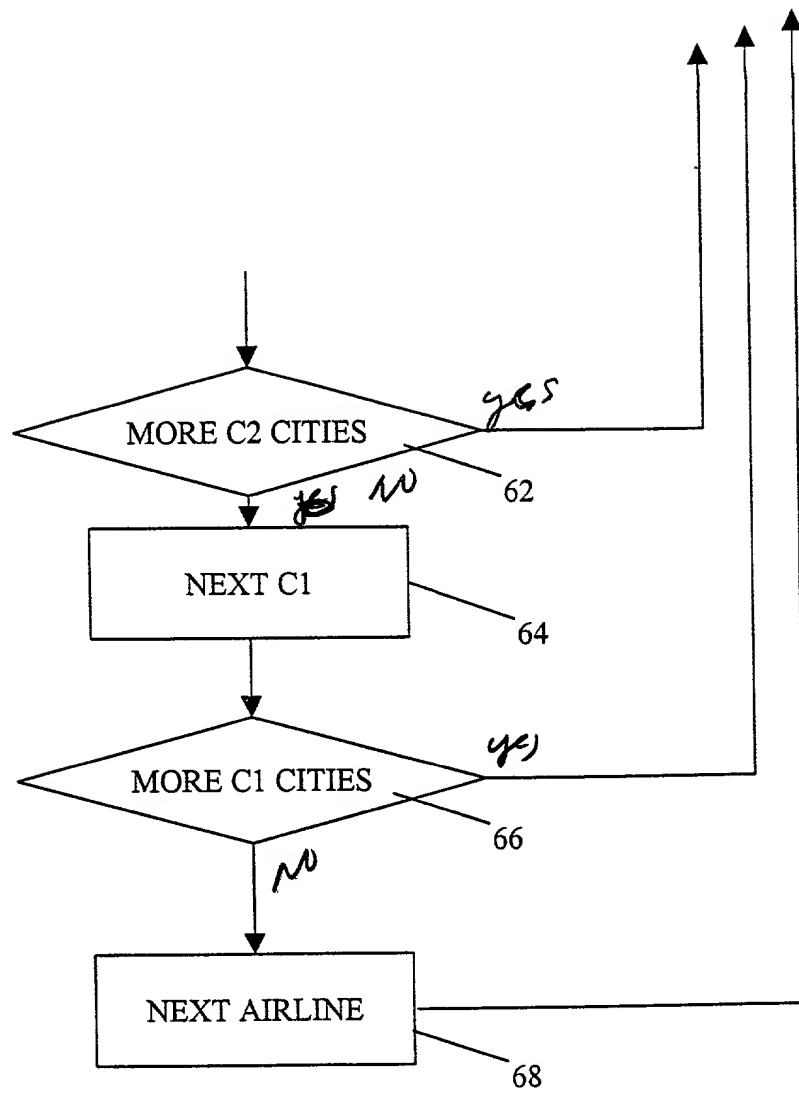


FIG. 5

